



PACKAGE CREATION

██████ UPDATES TO THE DISTRIBUTION DOCUMENT

AUTHOR

Michelle Turcotte

DATE MODIFIED

November 9, 2018

VERSION

1.1

© 2018 National Instruments Corporation ██████████

Do not reproduce without permission

OVERVIEW

PROBLEM DESCRIPTION

The current distribution document allows for package creation via a non-linear workflow, which makes it easy for users to miss key steps (like dependency calculation). No feedback is given on required information prior to building the distribution, at which point the build fails with errors.

This design aims to provide a simplified package creation flow which clearly informs users of required steps and fields, using existing information (such as user account) when available to reduce the number of required fields.

USER GOALS

Deliver Content

The user has created an application, library, or other content type they would like to share or deploy. To do this, they will likely need to create a package with all the necessary files, a list of dependencies, and information.

Install Content & Dependencies

The user may also want to get both their content as well as its dependencies installed on another machine. The user may also want a smaller-sized installer for portability, in which case the ability to add packages to feeds may be necessary.

Reuse

The user may wish to save the package configuration for reuse.

DESIGN PRINCIPLES

Promote Confidence

Encourage exploration and convey status, requirements, errors, and output behavior with visual cues and common interaction patterns. Use good defaults when possible.

Enable "Scrappiness"

Provide access to a non-linear workflow from one or more starting points for users familiar with package creation.

Progressive Disclosure

Promote good decision-making and reduce cognitive load through staged or progressive disclosure in the primary workflow.

INVOLVEMENT

Product Owner



Marketing Feature Owner



Interaction Design Lead

Michelle Turcotte

Visual Design Lead



Technical Lead



TABLE OF CONTENTS

REVISION HISTORY	4	INFORMATION SECTION	19
ACCESS POINTS	5	Default Values	
Menus		LICENSING SECTION	21
EMPTY DOCUMENT	7	Empty State	
Initial Empty State		Licenses Table	
Document Configuration		ERRORS & VALIDATION	24
FILES SECTION	10	Empty Fields	
No Files		Missing Files	
Files Table		APPENDIX	27
HARDWARE CONFIGURATION SECTION	13	Preview	
Empty State			
Hardware Table			
PACKAGE DEPENDENCIES SECTION	16		
Empty States			
Packages Table			

REVISION HISTORY

DATE	VERSION	UPDATES
11/9/18	1.1	Moved Hardware Configuration section above Package Dependencies Default display name changed from first document rename to default distribution document name ("Package") Defaults for Package name and Description changed to maintainer-projectname and Maintainer Project Name Licensing section changed to End User License Agreements (EULAs) (without support forLicense files)

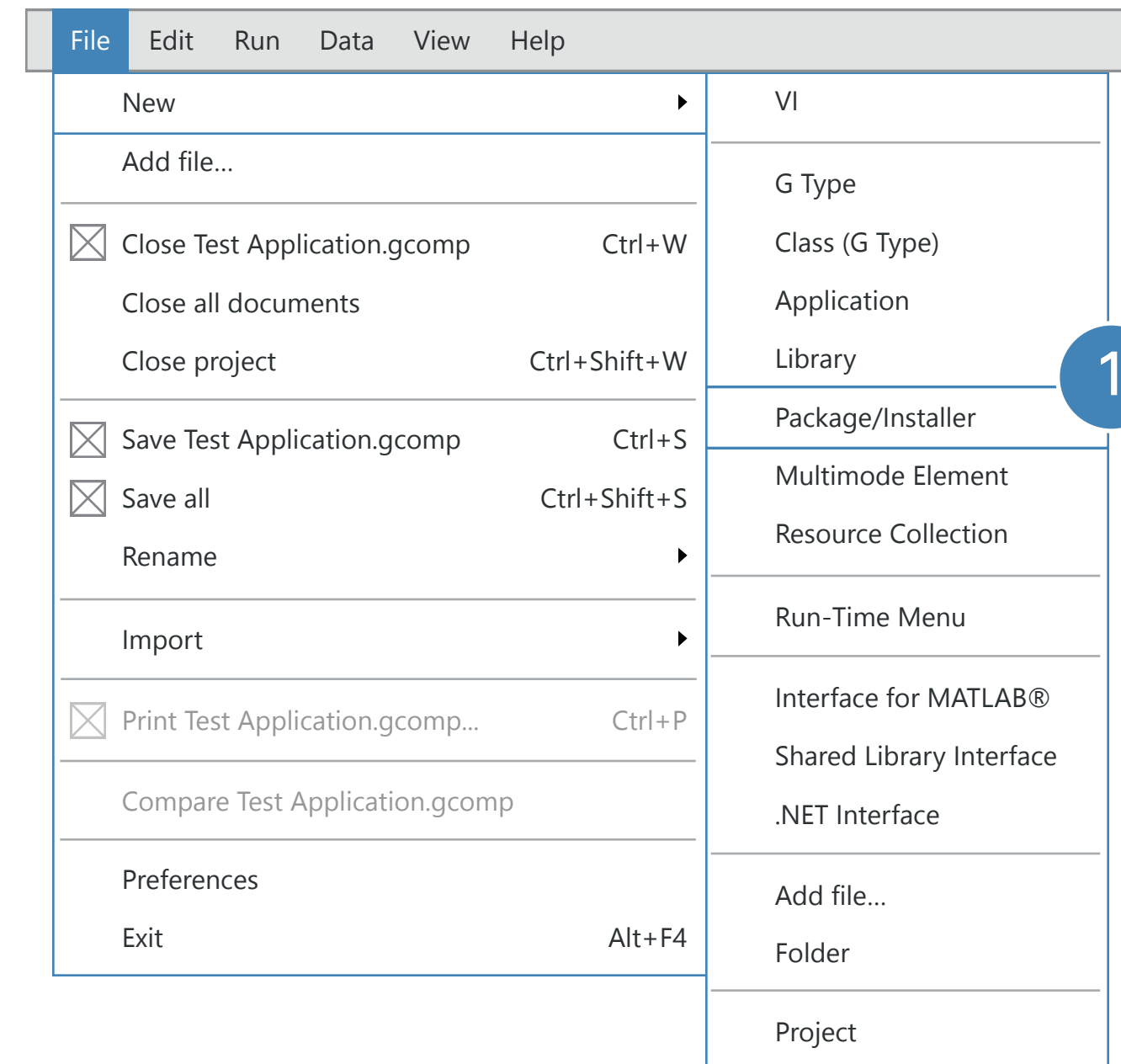
ACCESS POINTS

Various ways the user can begin the package creation workflow.

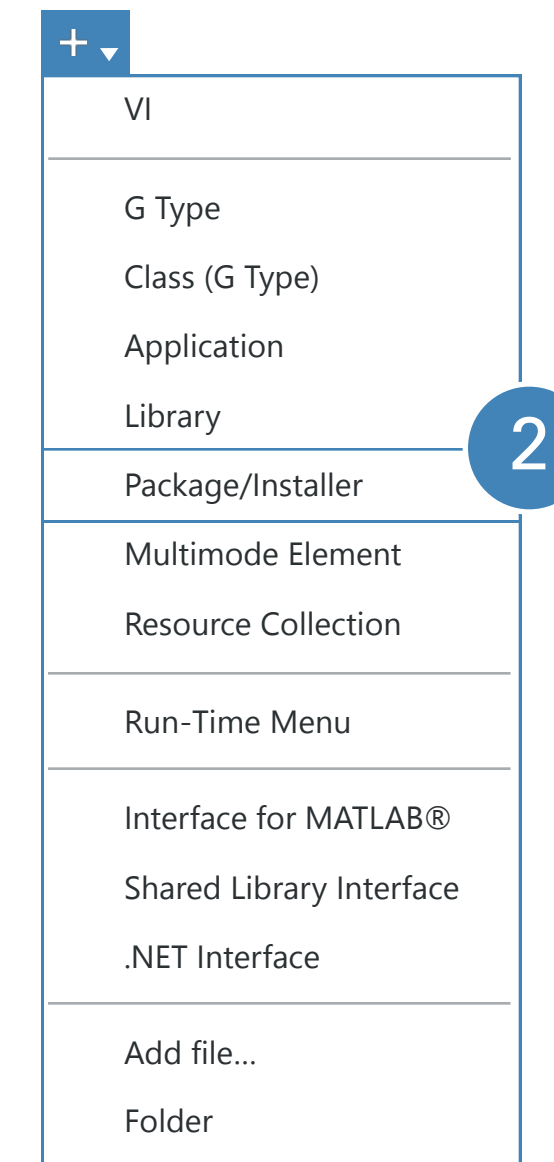
MENUS

- 1 **File → New**
Package/Installer option in File → New menu creates an empty .lvdist document.
- 2 **Files Pane New**
Package/Installer option in New creates an empty .lvdist document
- 3 **Document Tabs New**
Package/Installer option in New creates an empty .lvdist document
- 4 **Application/Library Context Menu**
Right click on an application or library contains an option to "Create Package/Installer with item". This creates a document that is populated with the application or library and its dependencies are captured automatically.
- 5 **Other Files Context Menu**
Right click on any other file type in the files pane contains an option to "Create Package/Installer with item". This creates a document that is populated with the file.

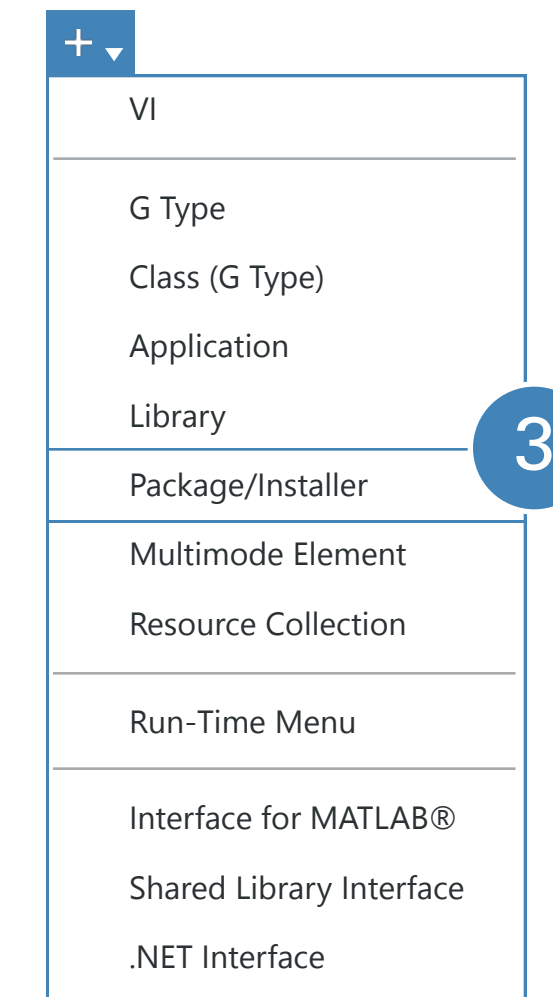
File Menu



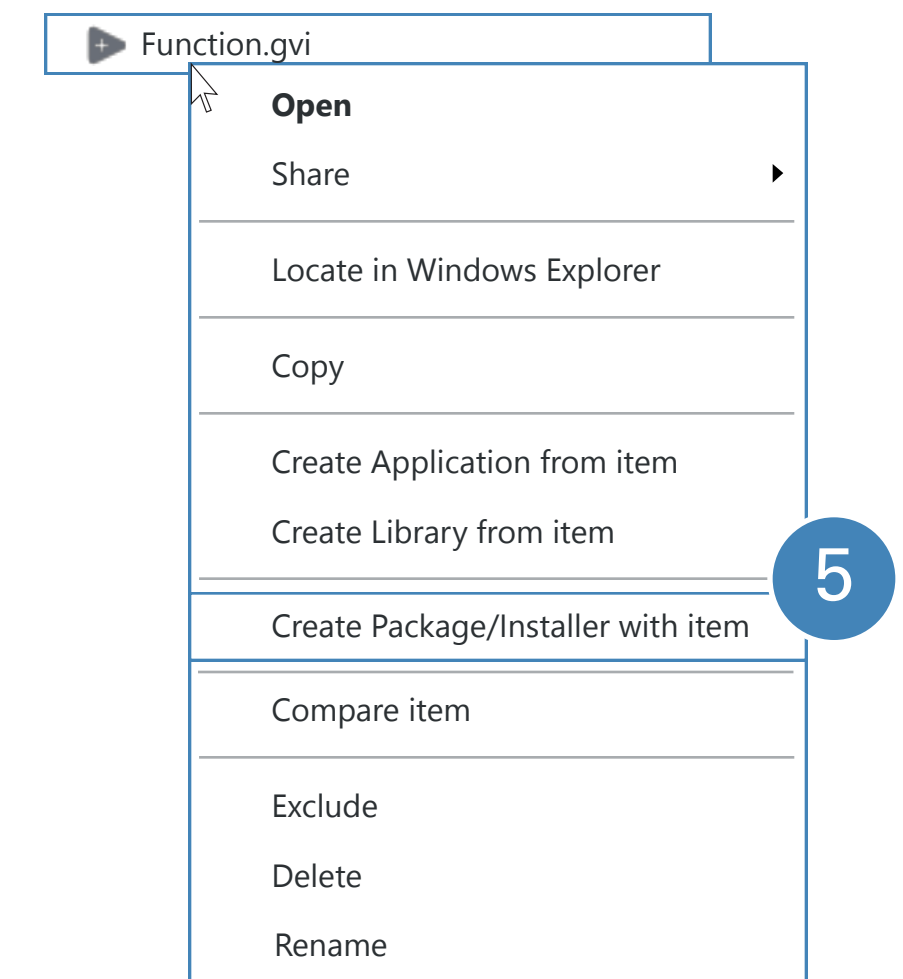
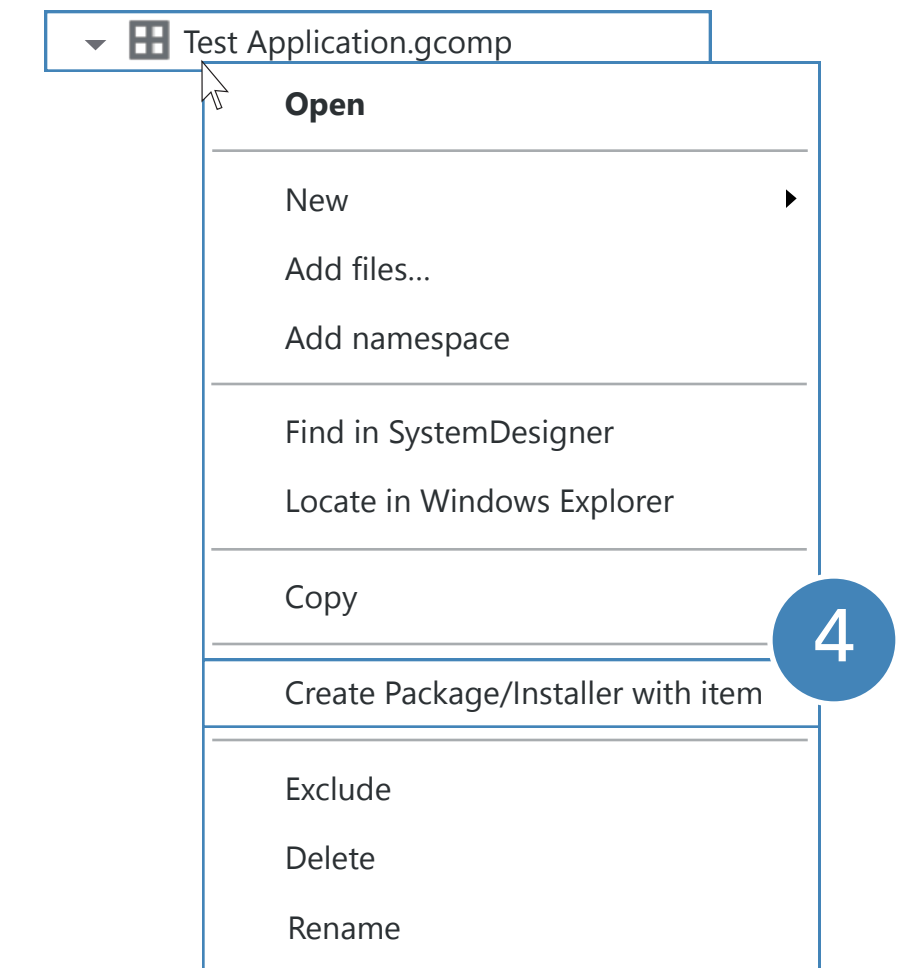
Files Pane



Document Tabs



File Context Menu

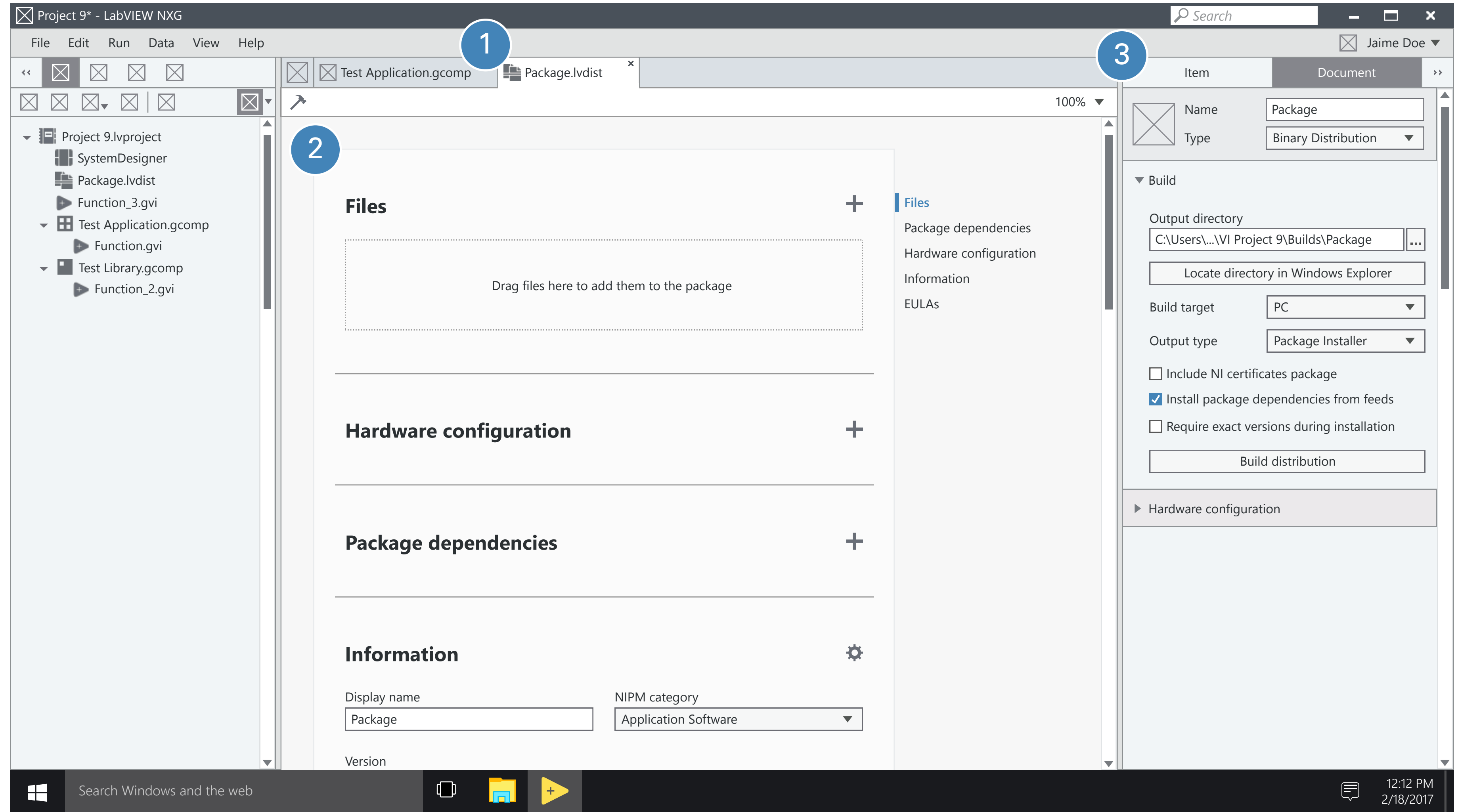


EMPTY DOCUMENT

The initial state of the Package/Installer document when no files have been added.


INITIAL EMPTY STATE

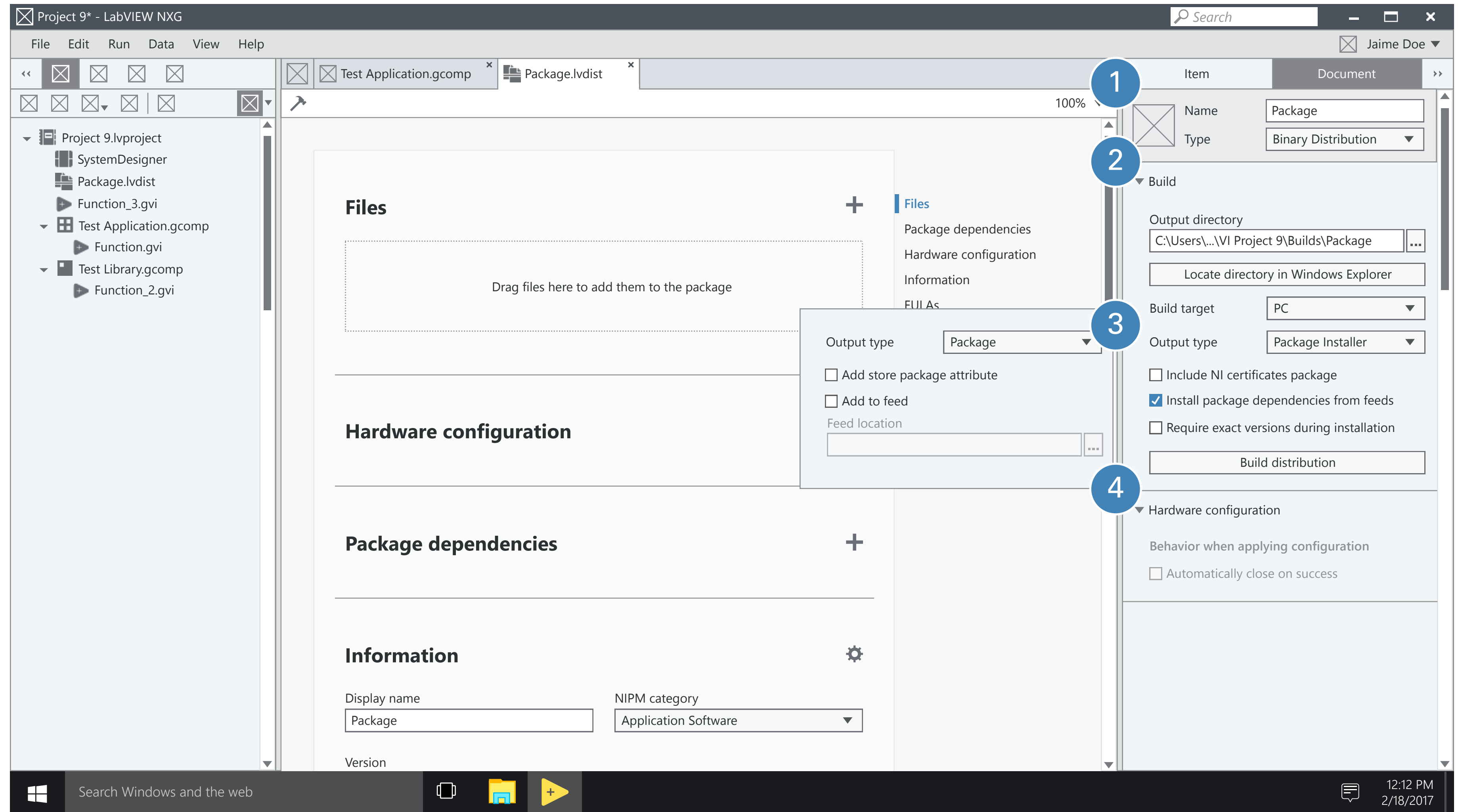
- 1 The .lvdist document created should default to the name "Package".
- 2 Sections should begin in their empty or default states on creation (see later sections for more details on Files, Hardware configuration, Package dependencies, Information, and Licensing sections).
- 3 The Document tab of the configuration pane should be open by default (and whenever there is no selection). The "Build" section should be open by default, and the "Hardware configuration" section should be closed by default.



DOCUMENT CONFIGURATION

The properties that are not moved into the Information section will remain largely the same in the document configuration pane, with some reordering for clarity.


- 1 The header remains the same (though the dropdown should have a border to match all other controls).
- 2 The remaining controls should be consolidated into the "Build" section. "Output directory" should be a full width path control with top-aligned label. "Build distribution" button should be moved to the bottom of the section for consistency with other documents.
- 3 The "Output type" dropdown remains as-is for . When "Package" is selected, all package-specific properties are displayed below, including Feed settings. When "Package Installer" is selected, properties below are installer-specific.
- 4 The "Hardware configuration" section remains as-is.

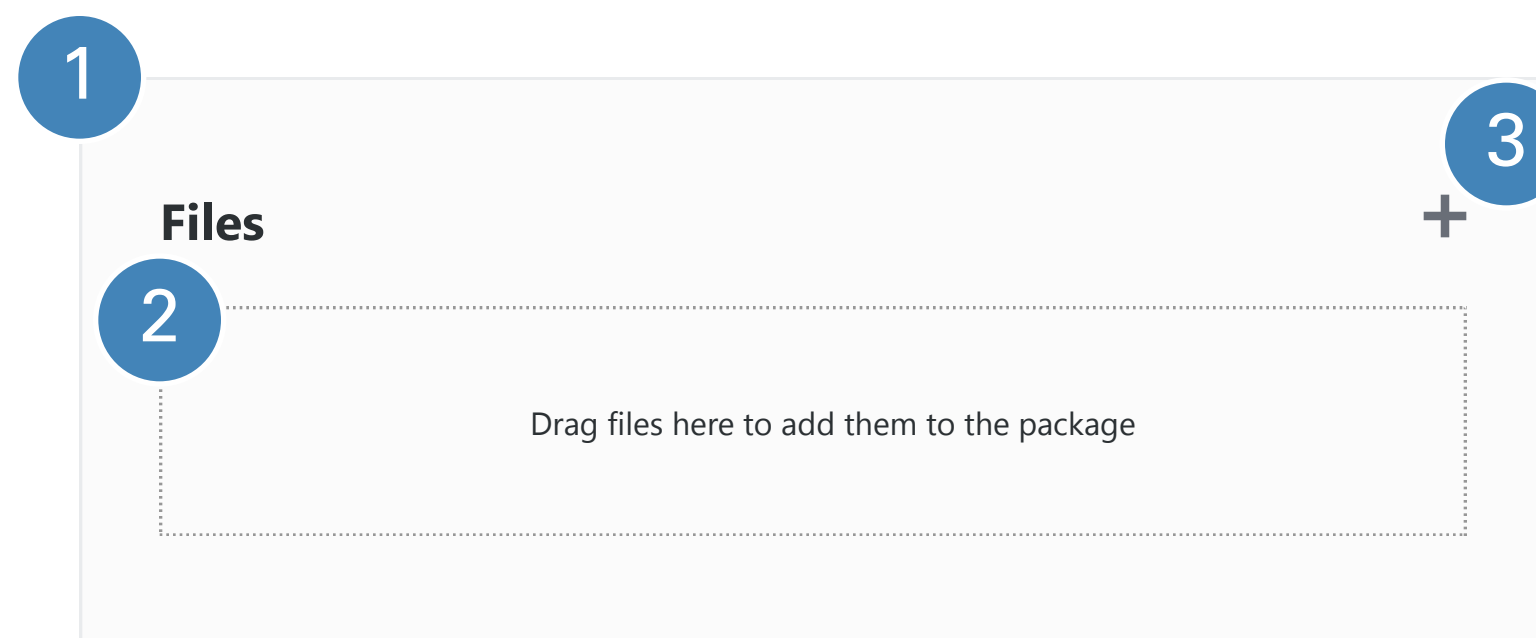


FILES SECTION

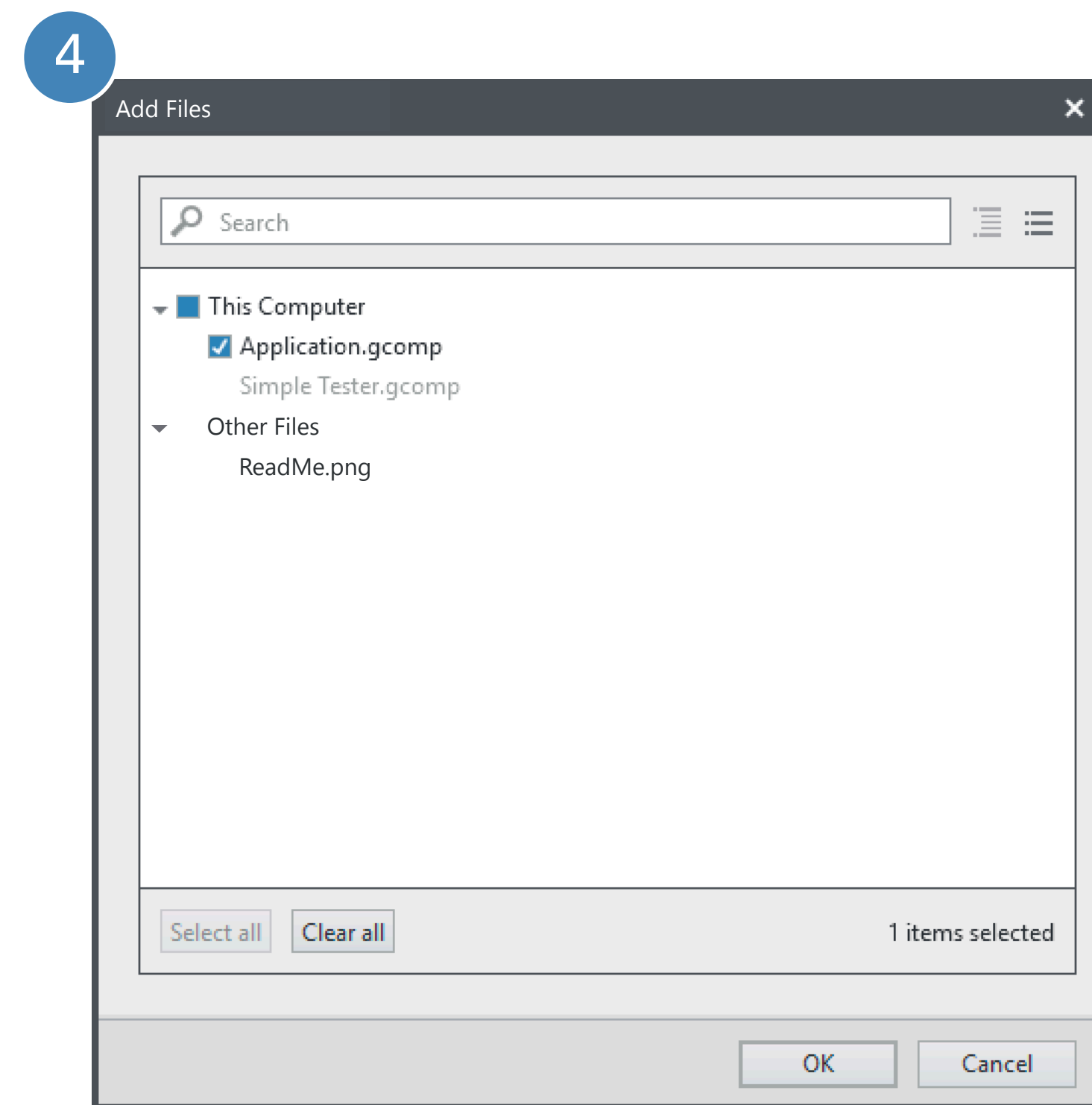
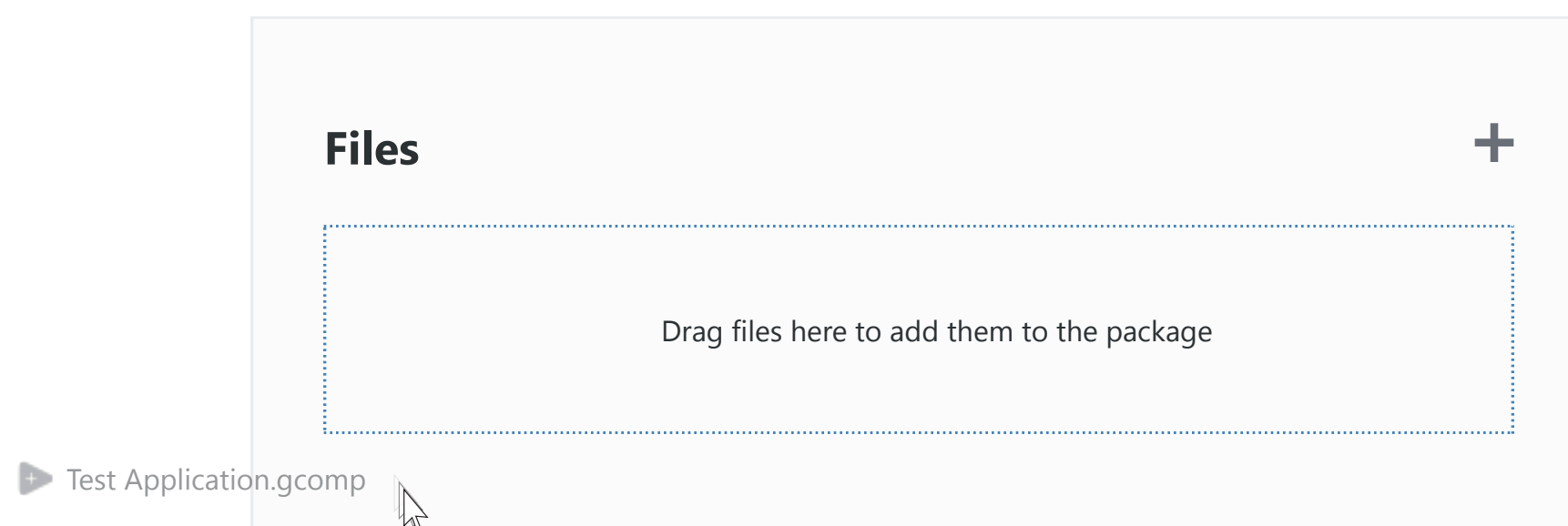
The various states, contents, and commands of the Files section in the document.

NO FILES


- 1 If the document is created in an empty state, or if all files are removed, the Files section should be in the empty state with the drag/drop target displayed.
- 2 When a file is dragged over the document space, the drag/drop target should change into a highlighted state to show where the file will be dropped in the document.
- 3 The "+" button should have a tooltip that says "Add files...". Clicking this should open the file selection dialog.
- 4 **File Selection Dialog**
For , this dialog should be a combination of the "Add Application/Library" dialog and the "Add Other Files" dialog, with a new dialog title of "Add Files". There should be two sections, one for the selected target and one for other files.



Drag Target Highlight State






FILES TABLE

- 1 When the files section has content, the drag/drop target should be hidden. Files can still be dragged in to the document at this point.
- 2 **Table Columns**
The table should have columns for Type (no label in header, icons only in this column); Name; Output; and Destination. Each of these columns should be sortable; Name should be the default sort column.
- 3 **Built Files (Apps & Libs)**
File types that are built before being added to the package (such as .gcomps) should be italicized with the text "build output" placed after the name. (Note that this added text should not affect sorting.)
- 4 **Output Dropdown**
If the file is an Application or Library, then the Output column should contain a dropdown with the following for : Rebuild and Last build. (Note that in the initial Stage, this will just be read-only text with the Rebuild option displayed.)
- 5 **Destination Flyout**
The flyout from Destination should duplicate the controls seen in the configuration pane. The flyout should close on click-away.

1



2

Files +

	NAME ▲	OUTPUT	DESTINATION
	ReadMe.txt		.../Public/Desktop
	<i>Test Application.gcomp build output</i>	Last build ▼	.../Public/Desktop
	<i>Test Library.gcomp build output</i>	Last build ▼	.../Public/Desktop

3

Files +

	NAME ▲	OUTPUT	DESTINATION
	ReadMe.txt		.../Public/Desktop
	<i>Test Application.gcomp build output</i>	Last build ▼	.../Public/Desktop
	<i>Test Library.gcomp build output</i>	Last build ▼	.../Public/Desktop

4

Last build ▼

Last build

Rebuild

5

.../Public/Desktop

Root location

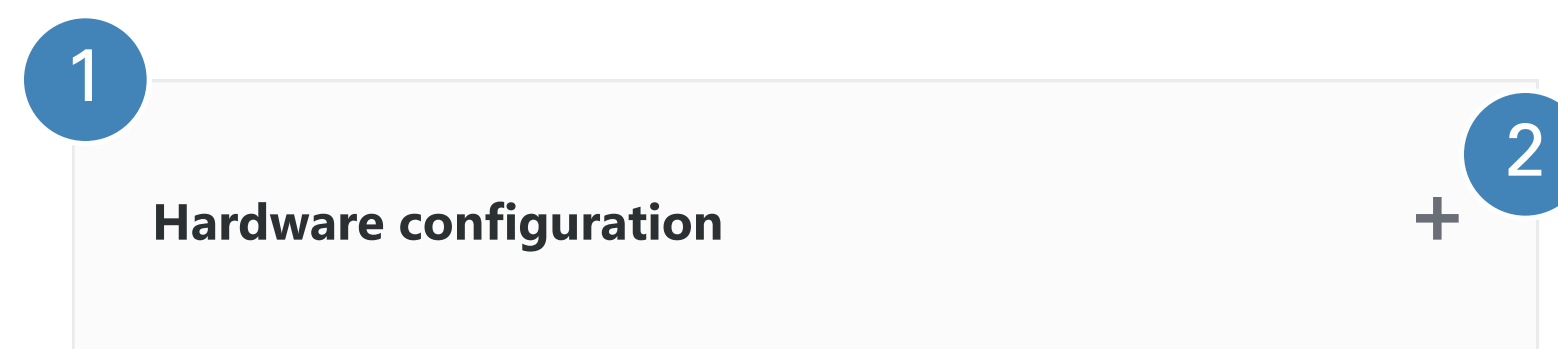
Relative location

HARDWARE CONFIGURATION SECTION

The various states, contents, and commands of the Hardware configuration section in the document.

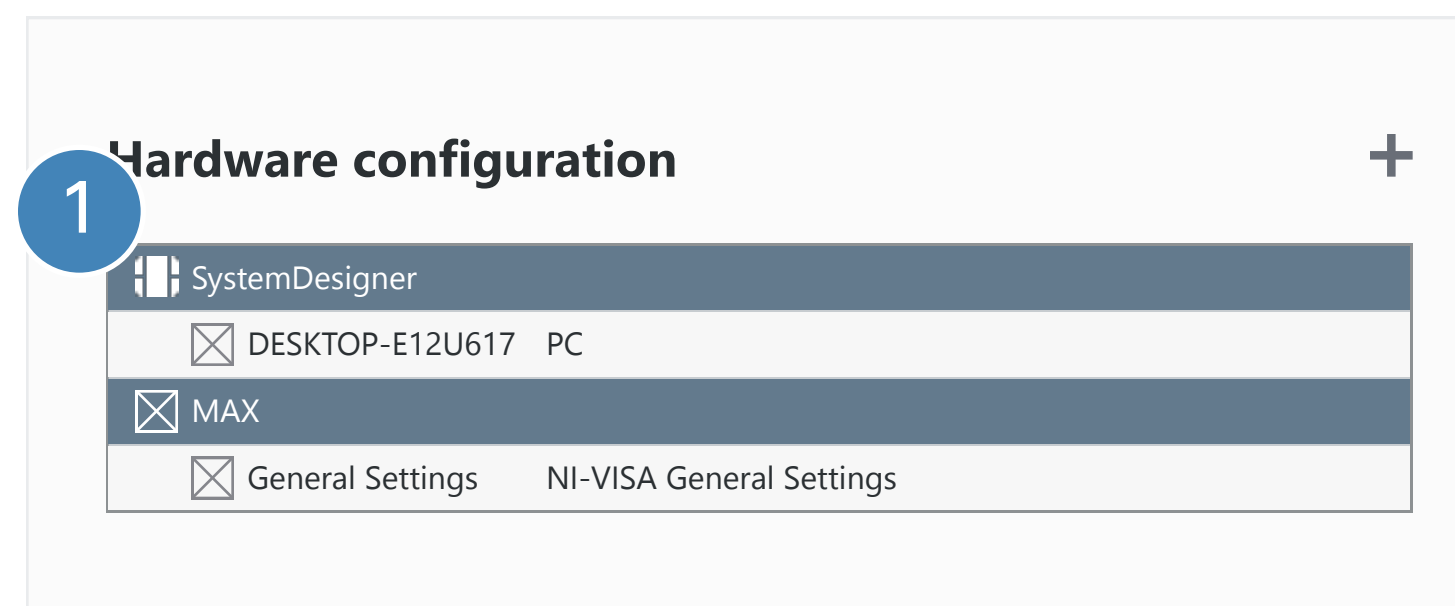
EMPTY STATE


- 1 The empty state of the Hardware configuration section should just have the header and the "+" button.
- 2 **Add Hardware Configuration**
The "+" button tooltip should say "Add hardware configuration..." and should open the existing hardware configuration dialog.



HARDWARE TABLE

- 1 The visual styling of the hardware configuration table should match the latest version of the Sortable Table Spec [REDACTED] and have sections and hierarchy as it does today.



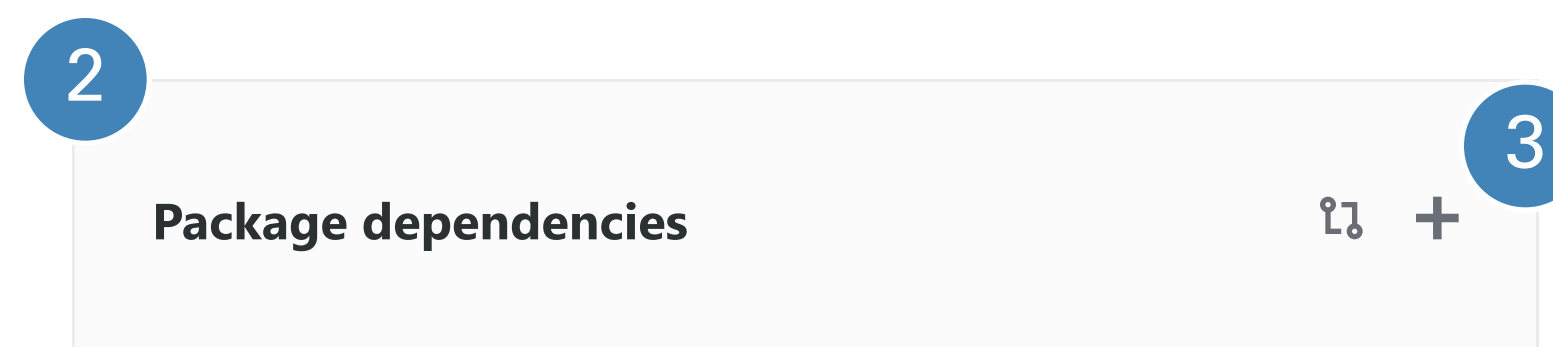
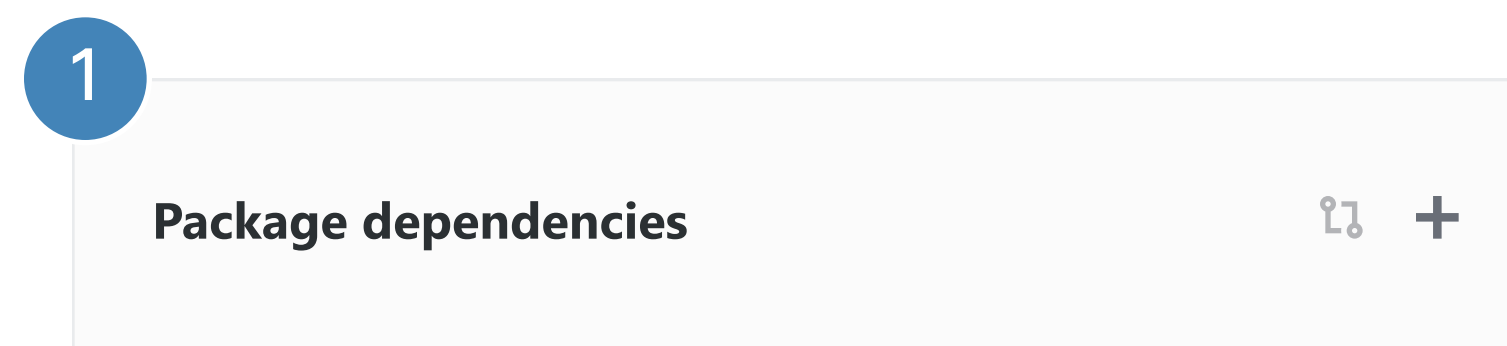
Hardware configuration		+
 SystemDesigner		
<input checked="" type="checkbox"/> DESKTOP-E12U617	PC	
<input checked="" type="checkbox"/> MAX		
<input checked="" type="checkbox"/> General Settings	NI-VISA General Settings	

PACKAGE DEPENDENCIES SECTION


The various states, contents, and commands of the Package dependencies section in the document.

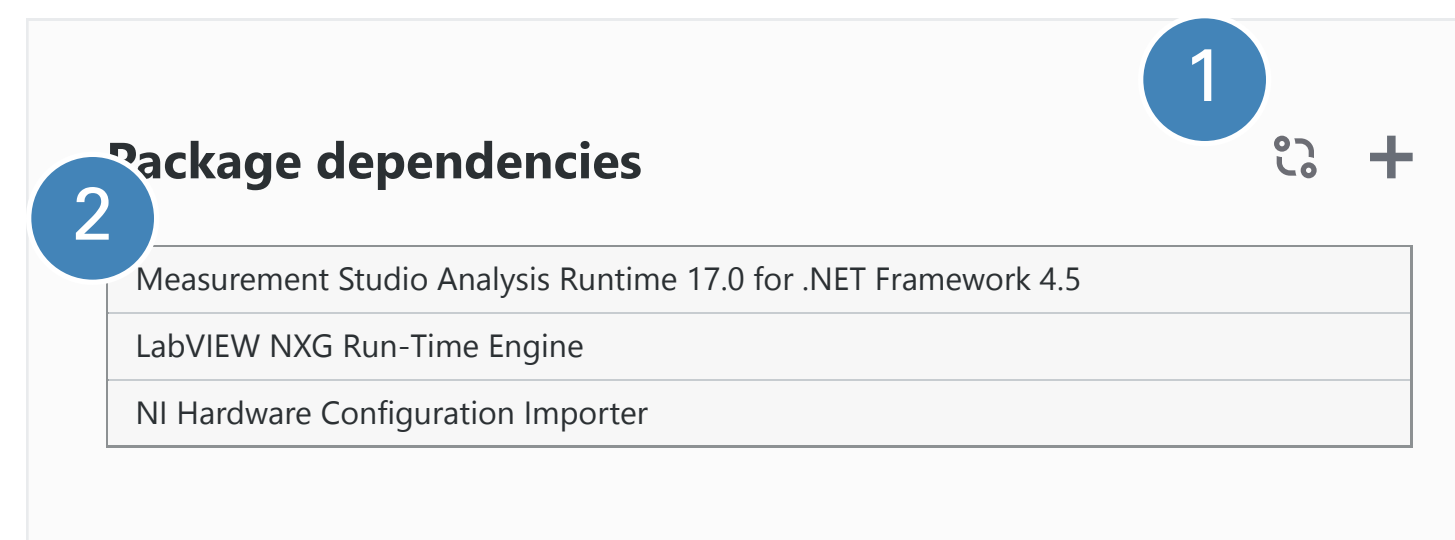
EMPTY STATES

- 1 No Files & No Hardware Config**
 If there are no files in the Files section and no hardware configuration added, then the empty state of the Package dependencies section should have the header and the "+" button, with the "Capture dependencies" button disabled.
- 2 Files/Hardware (No Dependencies)**
 If no dependencies have been found in the initial calculation, or the user cancelled that calculation or removed all dependencies, then a "Capture dependencies" button should be displayed when either files or hardware configuration exist.
- 3 Add Dependencies**
 The "+" button tooltip should say "Add package dependencies..." and should open the existing dependency dialog.



PACKAGES TABLE

- 1 Recapture Dependencies**
If package dependencies exist, then the "Capture dependencies" button should be updated to say "Recapture dependencies".
- 2** For , the SCT should be used to display the list of package dependencies, allowing for single- and multi-select. The configuration panes of these dependencies should remain the same as today.



INFORMATION SECTION

The various states, contents, and commands of the Information section in the document.

DEFAULT VALUES

- 1 Display Name**
The value of "Display name" should default to the default name of the .lvdist document (originally "Distribution", this will be changing to "Package").
- 2 NIPM Category**
The dropdown "NIPM category" should default to "Application software" for binary distributions.
- 3 Version**
The version should always start at 1.0.0.0, with auto-increment enabled.
- 4 Maintainer and Contact Email**
If the user is logged in, these fields should default to the values from the user's account. If not, then "Maintainer" should default to "Company" and "Contact email" should be left empty.
- 5 Package Name and Description**
When possible, package name is a combination of the value of "Maintainer" and the project name (lower case, no spaces or special characters). Description should be "Maintainer" and "Project name".

Information

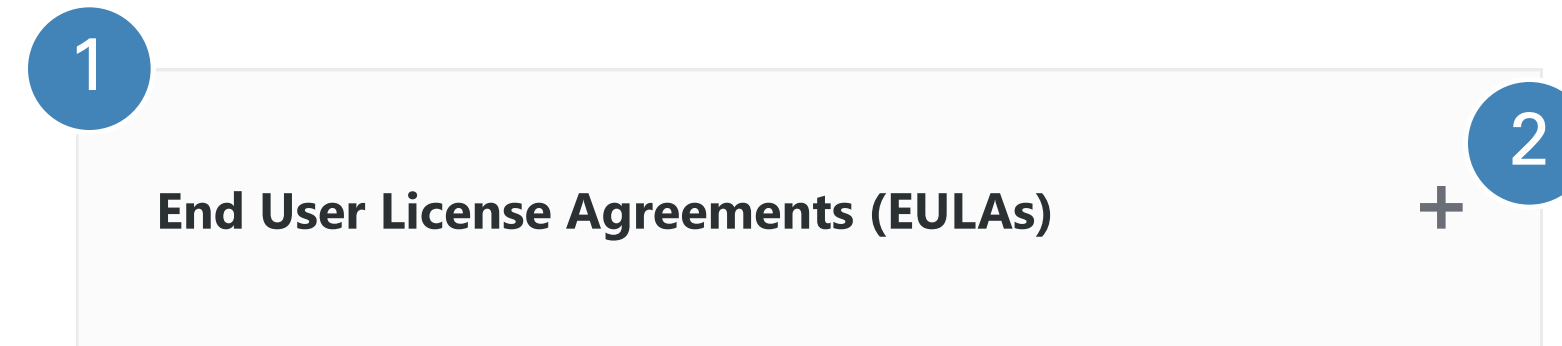
Display name <input type="text" value="Package"/>	NIPM category <input type="text" value="Application Software"/>
Version <input type="text" value="1"/> . <input type="text" value="0"/> . <input type="text" value="0"/> . <input type="text" value="0"/>	<input checked="" type="checkbox"/> Auto-increment version on build
Maintainer <input type="text" value="Acme Corporation"/>	Contact email <input type="text" value="user_account_email@acme.com"/>
Package name <input type="text" value="acmecorporation-projectname"/>	
Description <input type="text" value="Acme Corporation Project Name"/>	

END USER LICENSE AGREEMENTS SECTION


The various states, contents, and commands of the EULAs section in the document.

EMPTY STATE

- 1 The empty state of the Licensing section should just have the header and the "+" button.
- 2 **Add Licensing**
The "+" button tooltip should say "Add EULAs..." and should open the a browse dialog that filtered to EULA (.rtf) types.



LICENSES TABLE

- 1 The visual styling of the end user licenses list should match the latest version of the Sortable Table Spec  and will strictly list EULAs for the time being.




End User License Agreements (EULAs) +	
EULA_1.rtf	
EULA_2.rtf	

ERRORS & VALIDATION


Displaying errors and warnings in the document.

EMPTY FIELDS

- 1 In , if a field in the Information section is empty, it will have an error on it. This error should be listed in the Errors & Warnings pane, and should break the build button on the document.
- 2 Fields with errors on them should change border color to NLErrorBrush, and an icon (TBD) should be shown inside of the field on the far right. Double clicking on this icon should take the user to the error in the Errors and Warnings pane, and double clicking on the error in the pane should sonar highlight the field.


Information

1

Display name  NIPM category



Version . . . Auto-increment version on build

Maintainer Contact email




Package name  2

Description

MISSING FILES


1 Files with errors (such as if they are missing on disk) should follow the latest version of the Sortable Table Spec   for showing inline errors. Double clicking on the error icon should show the error in the Errors and Warnings pane, and double clicking in the pane should sonar highlight the file in the document.

Files +

	NAME	OUTPUT	DESTINATION
1 	ReadMe.txt		.../Public/Desktop
	<i>Test Application.gcomp build output</i>	Last build ▼	.../Public/Desktop
	<i>Test Library.gcomp build output</i>	Last build ▼	.../Public/Desktop

APPENDIX

PREVIEW

 We will later add a preview of the output of the package.

